(2)

AD-A246 644

FAULT TOLERANT REAL-TIME SYSTEMS

SFRC. N00014-84-K-0734

YEARLY REPORT

1 October 1990 - 30 September 1991

Prepared for:

Department of the Navy
Office of Naval Research
800 North Quincy Street
Arlington, Virginia 22217-5000

Prepared By:

John P. Lehoczky, Principal Investigator
Department of Statistics
Carnegie Mellon University
Pittsburgh, PA 15213
(412) 268-8725

Principal Investigator Names:

J. Lehoczky	(412) 268-8725	jpl@k.cs.cmu.edu
L. Sha	(412) 268-5875	lrs@sei.cmu.edu
D. Siewiorek	(412) 268-2570	dps@a.cs.cmu.edu
J. Strosnider	(412) 268-6927	jks@usa.ece.cmu.edu
H. Tokuda	(412) 268-7672	hxt@k.cs.cmu.edu

PI Institution: Carnegie Mellon University
Contract title: Fault Tolerant Real-Time Systems
Contract Number: N00014-84-K-0734
Reporting Period: 1 Oct 90 - 30 Sep 91

1 Productivity Measures

- Papers submitted but not yet accepted: 4
- Refereed papers accepted and in press: 4
- Refereed papers published: 17
- Books submitted or published: none
- Other reports: 8
- Ph.D. dissertations: 1
- Patents filed or granted: none
- Invited presentations: 10
- Contributed presentations: 20
- Honors, Prizes and Awards received:
  - **John Lehoczky**:
    - Member of the Board of Mathematical Sciences Panel on the High Performance Computer and Communications Initiative,
    - Chair, ASA Law and Justice Statistics Committee,
    - Member, NIH Special Study Section on Statistics,
    - Associate Editor, Journal of *Real-Time Systems*,
    - Member of the program committee of the 11th and 12th IEEE Real-Time Systems Symposium, the 8th IEEE Real-Time Operating Systems Workshop, the 11th ICDCS, and the 1992 SIGMETRICS and Performance 92 conferences.
  - **Lui Sha**
    - Chairman of the Board of Visitors of RICIS, an R&D center established by NASA and NASA JSC at University of Houston at Clearlake.
    - Program chair, 12th Real-Time Systems Symposium
  - **Hide Tokuda**
    - Chairman of the 8th Real-Time Operating Systems Workshop.
- Graduate students supported: 4
- Post-docs supported: 1
- Minorities supported: 2 graduate students and 1 post-doc

Principal Investigator Names:

J. Lehoczky	(412) 268-8725	jpl@k.cs.cmu.edu
L. Sha	(412) 268-5875	lrs@sei.cmu.edu
D. Siewiorek	(412) 268-2570	dps@a.cs.cmu.edu
J. Strosnider	(412) 268-6927	jks@usa.ece.cmu.edu
H. Tokuda	(412) 268-7672	hxt@k.cs.cmu.edu

PI Institution: Carnegie Mellon University
Contract title: Fault Tolerant Real-Time Systems
Contract Number: N00014-84-K-0734
Reporting Period: 1 Oct 90 - 30 Sep 91

2 Summary of Technical Progress

2.1 Overview

The ART (Advanced Real-Time Technology) Project of Carnegie Mellon University is engaged in wide ranging research on hard real-time systems. The project has as its overall goal the development and demonstration of predictable and fault tolerant hard real-time computer systems. To achieve this goal, research is being conducted in three interrelated areas:

1. The development of a theory of hard real-time resource management including scheduling and synchronization. In addition, this theory is being applied in a wide range of contexts including databases, communication networks, operating systems and artificial intelligence.
2. The design and construction of an operating system (ARTS) that supports predictable and fault tolerant real-time computer systems and demonstration and testing in the ARTS testbed to provide proof of concept and to gain understanding into aspects of the theory needing further refinement. A real-time version of the Mach operating system, RT-Mach, is also being developed as a part of the project.
3. Development of hardware architectures and approaches to fault tolerance that can support predictable hard real-time computer systems.

The project members are also concerned with the transition of the theory being developed to the user community, especially the U.S. Navy. For this reason, project members have a close relationship with the RMARTS project (formerly RTSIA) in the CMU Software Engineering Institute. Dr. Lui Sha is a member of both the ART Project and the RMARTS project and has made great strides in increasing adoption of the rate monotonic scheduling theory in major national projects.

During the October 1, 1990 - September 30, 1991 period, substantial progress was made in each of these broad categories. That progress is briefly discussed below. A more detailed collection of briefing materials is contained in the *ART Project Review Briefing* which was given to the ONR representative.

2.2 Theory of Hard Real-Time Resource Management

The theoretical research is focused on fixed priority scheduling algorithms which include the rate monotonic and deadline monotonic algorithms. This family of algorithms is designed to handle in a predictable fashion many important real-time system characteristics including: a mixture of hard-deadline periodic, hard deadline aperiodic and soft deadline aperiodic tasks, task synchronization and scheduling distinct resources including processors and communication media.

Significant advances were made in scheduling theory and in the application of this theory to a range of real-time systems problems. These advances include:

- New results on scheduling periodic tasks which are composed of serially executed subtasks.
- A new protocol, the read_write ceiling protocol for improving the performance of hard real-time databases.
- Results on end-to-end scheduling, especially guaranteeing periodic tasks which are executed on a distributed system with end-to-end deadlines. A new scheduling algorithm was introduced to handle the jitter problem that can arise with I/O communication and distributed systems.
- The scheduling theory was extended to evaluate the overhead associated with scheduling and context swapping for specific operating systems and hardware architectures.
- Scheduling models were extended to handle communication networks including models for IEEE 802.6.
- Scheduling theory was used, in the form of aperiodic servers, to provide predictable real-time AI systems.

2.2.1 Scheduling Complex Tasks

This research involves the fixed priority scheduling of tasks with varying execution priorities. In this research, a set of periodic tasks, each of which is composed of a set of serially executed subtasks, is considered. Each subtask has a computation requirement, a deadline and a fixed execution priority. This formulation allows us to extend the fixed priority schedulability conditions to consider both preemptive and nonpreemptive scheduling as well as tasks with precedence constraints. An exact schedulability condition was developed. It is implemented by checking the timing requirements of each task. This is accomplished by reducing each task to "canonical form," categorizing the remaining tasks into one of five types and checking a set of detailed scheduling conditions. This is accompanied by a theoretical analysis which generalizes the Liu and Layland bounds.

2.2.2 End-to-end Scheduling

The goal is to extend the rate monotonic scheduling theory to distributed systems. The end-to-end schedulability problem arises when periodic tasks must use a sequence of schedulable processing elements (e.g. processors and communication media) to complete and have a hard deadline timing requirement only on the task completion time. If one were to allow a single period of latency for each schedulable entity, then the rate monotonic algorithm could be used at each schedulable entity, and our current results could be applied to give an analysis of both the worst and average cases. For some systems, the resulting latency may be too long, and one must shorten the time to completion. Our initial approach has been to apply fixed priority scheduling methods in each resource in such a way that the end-to-end deadlines are met. It is not necessary to divide the end-to-end deadlines evenly at each resource, and tasks can be given longer intermediate deadlines at resources which are heavily utilized. Task deadlines may no longer agree with task periods, so a modified fixed priority scheduling algorithm such as the deadline monotonic algorithm must be used. Moreover, this approach may result in two tasks having different priorities in different resources.

During this contract period, the jitter problem was addressed. In the context of distributed system scheduling, jitter occurs when a job arrives either earlier than scheduled or later. In the case of an early arrival, the job may cause a lower priority job to miss its deadline. In the case of a late arrival, the job may miss its own deadline. To overcome this problem, we developed a new fixed priority algorithm, the maximally jittered deadline monotonic algorithm, and analyzed its performance.

2.2.3 Schedulability Analysis of Specific Operating Systems

Over the past year we developed analysis techniques for micro-kernels, which focused on the three core elements of modern micro-kernels: task management, synchronization, and communications. Focus of much of this year's work was on task management (scheduling). Models were developed for several basic scheduling mechanisms, including both timer and event driven scheduling. These mechanisms were analyzed for sources of overhead and blocking in the areas of interrupt/trap handling, scheduling code, and context switching. This analysis covered features of modern microprocessors that affected the overhead/blocking components.

The rate monotonic scheduling equations were modified to account for each basic scheduling mechanism. The modified equations account for the overhead and blocking identified above. An analysis tool was developed to scale arbitrary task sets to their breakdown utilization based on a particular scheduling implementation, using particular overhead values. Using this tool, the implementations can be compared for the relative performance in terms of schedulable utilization. Other results include a method for determining optimal clocking frequency for a given task set/implementation, and a method of quantitatively comparing processors for real-time applications, again using breakdown utilization as a meter.

2.2.4 Integrated Real-Time AI

Real-time mission-critical systems present unique challenges for AI applications. The primary problem with AI tasks in real-time environments is that the worst case execution time is often unknown, or is many orders of magnitude larger than the average case. When the run-time variance is large, the schedulable utilization of the resource is very low, resulting in inefficient implementations. Further, if the execution time of AI tasks is not constrained, they can affect the timeliness of the other real-time tasks in the system.

The sources of the execution time variations of the AI tasks were examined in the context of the Newell/Simon problem space hypothesis. We then developed a real-time problem solving model which is based on the fundamental degrees of freedom available in manipulating a problem space. The model is used to map existing techniques to a composition of these fundamental operations. This provides a framework to reason about the effect of these operations on the worst case and the average case execution time of the problem solving task.

To maintain predictability of other real-time tasks in an integrated system, we have developed encapsulation methodologies within an AI Server, and a real-time production system(CROPS5) as an instantiation of this approach. Two complex, real-world applications are used as testbeds to evaluate the effectiveness and limitations of this approach.

2.3 The ARTS Real-Time Operating System

ARTS 2.0 is an object oriented real-time kernel with many important features supporting predictable hard real-time computer systems. The kernel features an object oriented thread model with both periodic and aperiodic threads. It has an integrated time driven scheduler with policy and mechanism separation which allows the user to select a wide range of policies (for example, rate or deadline monotonic, deferrable or sporadic servers). The kernel fully supports real-time synchronization policies such as the priority ceiling protocol and real-time communications including RTP, VMTP, RTCN, XTP and UDP protocols. The kernel also has real-time file management. Finally, there is an associated toolset: Scheduler 1-2-3 for predicting task set schedulability and ARM 2.0 for real-time system monitoring.

2.4 Real-Time Architecture and Fault Tolerance

2.4.1 Real-Time RISC Architecture

This effort developed a high performance, real-time RISC pipeline architecture that subsumes scheduling and event handling in pipeline stalls with little additional hardware. The resulting architecture is the product of a quantitative, scheduling theoretic approach whose goal was to narrow the gap between real-time scheduling theory and its implementation in operating systems running on RISC processors. We first developed a scheduling model which highlights the overhead components associated with conventional software implemented scheduling and event handling. We then investigated the viability of adding a scheduling/event handling engine which supports parallel execution of operating system functions and application functions. A scheduling theoretic model of this approach was developed, and it is shown that state management functions must still operate serially on the application processor resulting in substantial visible overhead. Although this approach provides a much more direct implementation of real-time scheduling theory, the additional hardware required is difficult to justify. Noting that RISC pipelines are typically 10 to 15 percent idle due to pipeline stalls, we then investigated the viability of converting these stall cycles into a predictable scheduling/event handling engine. Not only did we demonstrate that these stall cycles can be converted into a deterministic operating system engine, but we also developed parallelizable state management functions which were previously unparallelizable. This technique uses pipeline stall cycles to maintain the state of the next task to be executed thus providing a seamless context swap capability. This capability along the ability to perform scheduling and event handling in the pipeline stalls results in a RISC pipeline architecture that cleanly implements real-time scheduling theory nearly devoid of all implementation artifacts. Further, by subsuming substantial operating system overhead components within pipeline stalls, the same RISC pipeline in effect has a 10 to 15 percent performance boost.

The challenge was thus to demonstrate that such a high performance, real-time RISC architecture is feasible with limited additional hardware. The new architecture is in affect a multithreaded architecture where the application processing is one thread and the operating system processing is a second thread. We show that our specialized operating system thread compliments as opposed to competes for processor resources, and thus eliminates most of the difficulties associated with realizing generalized multithreaded architectures. We show that our real-time RISC architecture can be implemented with less than 15% additional hardware without significant degradation in processor speed. The design is then extended to super-pipelined and super-scalar RISC organizations as well where there are even more opportunities to support operating system processing in pipeline stalls.

2.4.2 Real-Time Networks

During the past year we have made significant progress in each of the following networking areas as summarized below.

- Scheduling Model for ATM-BISDN Switches: Developed a model for the Sunshine Switch developed by Bellcore that develops worst case bounds for delay through the switch. In particular the model considers the case when several inputs are directed at the same output and develops bounds on the delay, the maximum required output buffers and maximum required recirculating buffers.

- Schedulable Routing Algorithm: Developed an algorithm to route traffic in an arbitrary topology multihop network, with a connection-oriented network service, such that end-to-end timing guarantees can be made. The algorithm calculates the degree of schedulability saturation of each link in the network and selects a path between source and destination that traverses the least saturated link. This keeps the load in the network balanced and minimizes the possibility of saturating any link.
- Scheduling Analysis of Dual Link Metropolitan Networks: Developed a theoretical framework to analyze schedulability of dual link MANs such as IEEE 802.6 (DQDB). Showed that current MAN protocols exhibit unpredictable behavior. The fundamental challenge here is to ensure predictable operation in spite of incomplete information in both time and space. To address the unpredictability problem we introduced the concept of system coherence. A system is coherent if the distributed queues in each station in the network are consistent with some observable ordering of requests. We derive the conditions for system coherence and develop a scheduling model that allows calculation of end-to-end message delay between a source and destination.
- XTP/FDDI Scheduling Analysis: Recently kicked-off a project that will develop scheduling analysis of the FDDI Data link layer controlled by the XTP transfer protocol. We aim to develop models that will allow determination of medium access delays for XTP clients. This will require joint analysis of XTP. We will leverage our existing expertise in IEEE 802.5 token ring analysis to develop a model for FDDI.

2.4.3 Real-Time Memory Hierarchies

This past year saw the completion of our research on predictable cache architectures for real-time systems. Additionally we developed a generalized dynamic-programming algorithm for optimally *partitioning two-level memory systems for* deterministic, real-time performance. Further, it supports on-line reconfiguration of memory resources in response to changing requirements because it runs in polynomial time. This capability is crucial to the next generation of adaptive, real-time systems. We found that two classes of schedulability tests - utilization bound and iterative tests - can be incorporated into the algorithm to ensure that task deadlines will be met.

Future plans for this system include expanding the types of objects, incorporating more extensive rules to allow object determination, running the system under a real-time system based on RT Mach, and to use the system to investigate integrating learning paradigms in real-time systems.

2.4.4 FAA Prototype

An initial distributed FAA Prototype was developed this year which simulates aircraft and in which some of the planes can contain integrated real-time artificial intelligence capabilities that control their behaviors. The first part of this project integrated the CROPS5 real-time production system, a lightweight process package, a remote procedure call package, the X window system, and the Unix timer facility. These components were integrated to create a coincidently real-time, distributed collision avoidance demonstration system running under Ultrix. Under this system, there is a single server process which maintains the information about the objects in the simulation. There are some client processes which generate "unintelligent" planes, which are planes on random, straight trajectories. The simulation may also contain "player" processes, in which a human being controls a plane from his workstation and can watch the progress of the other planes. The other clients are the "intelligent" planes which are under the control of a production system written in CROPS5. The production system currently detects when planes enter the "red zone" around its plane and attempts to change the course of the plane to avoid hitting any other planes.

2.4.5 Intelligent Controls Testbed

In the Dynamic Scheduling Pacing System testbed, an initial schedule is formulated based on steel orders from the MRP and the time constraints of the caster. This schedule is implemented on the process by lower level controllers and the operator. A behavioral model monitors the inputs and outputs of the steel process over time. The monitor generates an encoding of the possible process evolutions corresponding to the observations and the expected variation within the system. When this encoding becomes empty, i.e. when the system has deviated beyond its expected operations, the monitor generates an alarm indicating that rescheduling may be needed.

Upon receiving this process deviation alarm, the intelligent reactive scheduler tries to localize the source of deviation with the aid of a human operator. In cases where the required resources are not available, the reactive scheduler determines the constraints that can be relaxed, and a new control plan is generated. This new plan is then checked for validity using a spawned version of the behavioral model in a simulation mode. The steps of the new control plan are implemented on this simulation to determine if the recovery actions will retain the temporal and metallurgical properties desired. If there are violations, the iteration between the reactive scheduler, a linear programming package and the behavioral model is repeated, using rules to relax various constraints. Once an acceptable new control plan is determined, it is provided to the operator and lower level controllers.

The intelligent reactive scheduler and the behavioral model monitor are currently being integrated and evaluated. The behavioral model monitor and the reactive scheduler operate concurrently on a DecStation 5000/200, under the Mach operating system.

Principal Investigator Names:

J. Lehoczky	(412) 268-8725	jpl@k.cs.cmu.edu
L. Sha	(412) 268-5875	lrs@sei.cmu.edu
D. Siewiorek	(412) 268-2570	dps@a.cs.cmu.edu
J. Strosnider	(412) 268-6927	jks@usa.ece.cmu.edu
H. Tokuda	(412) 268-7672	hxt@k.cs.cmu.edu

PI Institution: Carnegie Mellon University
Contract title: Fault Tolerant Real-Time Systems
Contract Number: N00014-84-K-0734
Reporting Period: 1 Oct 90 - 30 Sep 91

3 Publications and Presentations

The following are project publications which were written, appeared or are in press for the above period. Project members made many research presentations during the contract period, and these are not enumerated. Each of the articles listed below that was published in a conference proceedings volume was presented at that conference.

3.1 Published or In Press

- Harbour, Michael Gonzalez, Klein, Mark H., Lehoczky, John P., "Fixed Priority Scheduling of Periodic Tasks with Varying Execution Priority", to appear in *Proceedings of the 12th IEEE Workshop on Real-Time Systems Symposium*, December 3-6, 1991.
- Ramos-Thuel, Sandra and Strosnider, Jay K., "The Transient Server Approach to Scheduling Time-Critical Recover Operations", to appear in *Proceedings of the 12th IEEE Workshop on Real-Time Systems Symposium*, December 3-6, 1991.
- Tokuda, Hideyuki and Nakajima, Tatsuo, "Evaluation of Real-Time Synchronization in Real-Time Mach", to appear in *Proceedings of 2nd Mach Symposium*, November 1991.
- Mercer, Clifford M. and Tokuda, Hideyuki, "Priority Consistency in Protocol Architectures", in *Proceedings of 16th IEEE Local Computer Networks Symposium*, October 1991
- Paul, C. J., Palmer, G., Strosnider, J. K., "Analysis of Architectures for Fault-Tolerant Computation", in *Communications of the ACM*, August 1991, Pages 80-93.
- Holloway, L. E., Paul, C. J., Strosnider, Jay K. and Krogh, B. H., "Integration of Behavioral Fault-Detection Models and an Intelligent Reactive Scheduler", in *Proceedings of Intelligent Controls Conference, Washington, DC*, August 1991.
- Sha, L., Rajkumar, R., Lehoczky, J., Son, S. and Chan, C.-H., "A Real-Time Locking Protocol", in *IEEE Transaction on Computers*, July, 1991.)
- Sha, L., Rajkumar, R. and Lehoczky, L, "Real-Time Computing Using Futurebus+", in *IEEE Micro*, June 1991.
- Lehoczky, J., Sha, L., Strosnider, J. and Tokuda, H., "Fixed Priority Scheduling Theory for Hard Real-Time Systems", in *Foundations of Real-Time Computing: Scheduling and Resource Management*, edited by van Tilborg, A. M. and Koob, G. M., *Kluwer Academic Publishers*, 1991.
- Sha, L., Klein, M. and Goodenough, J., "Rate Monotonic Analysis for Real-Time Systems", in *Foundations of Real-Time Computing: Scheduling and Resource Management*, edited by van Tilborg, A. M. and Koob, G. M., *Kluwer Academic Publishers*, 1991.
- Kirk, David B., Strosnider, Jay K. and Sasinowski, John E., "Allocating SMART Cache

Segments for Schedulability", in *Foundations of Real-Time Computing: Scheduling and Resource Management*, edited by van Tilborg, A. M. and Koob, G. M., *Kluwer Academic Publisher*, 1991.

- Volz, R., Sha, L. and Wilcox, D., "Maintaining Global Time in IEEE Futurebus+", in *Real-Time Systems Journal*, vol. 3, No. 1, 1991.
- Mraz, R., Strosnider, J. K., White, M., and Palmer., G., "Analysis of Architectures for Fault-Tolerant Computation", in *Proceedings of the 24th Hawaii International Conference on System Sciences*, Award Paper, January, 1991.
- Kirk, D. B., and Strosnider, J. K., "SMART (Strategic Memory Allocation for Real-Time) Cache Design Using the R3000", in *Proceedings of the 11th IEEE Real-Time Systems Symposium*, December, 1990.
- Sha, L., Rajkumar, R., Lehoczky, J., "Real-Time Scheduling Supports in Futurebus+", in *Proceedings of 11th IEEE Real-Time Systems Symposium*, December, 1990.
- Lehoczky, J., "Fixed Priority Scheduling of Periodic Tasks with Arbitrary Deadlines", in *Proceedings of 11th IEEE Real-Time Systems Symposium*, December, 1990.
- Mercer, C. W., and Tokuda, H., "The ARTS Real-Time Object Model", in *Proceedings of 11th IEEE Real-Time Systems Symposium*, December, 1990.
- Rashid, R. and Tokuda, H., "Mach: A System Software Kernel", *in Proceedings of Symposium on Computational Technology for Flight Vehicles*, November 1990.
- Tokuda, Hideyuki, Mercer, Clifford W. and Breach, Scott E., "The Impact of Priority Inversion on Continuous Media Applications", in *Proceedings of 1st International Workshop on Network and Operating System Support for Digital Audio and Video*, November 1990.
- Tokuda, H., Nakajima, T. and Prithvi Rao, "Real-Time Mach: Towards Predictable Real-Time Systems", in *Proceedings of USENIX 1990 Mach Workshop*, October 1990.
- Ishikawa, Y., Tokuda, H. and Mercer, C. W., "Object-Oriented Real-Time Language Design : Constructs for Timing Constraints", in *Proceedings of Joint ACM OOPSLA/ECOOP '90 Conference on Object-Oriented Programming: Systems, Languages, and Applications*, October 1990.

3.2 Submitted for Publication and Other Reports

- Rajkumar, R. and Sha, L., "A Real-Time Multiprocessor Locking Protocol for Database Applications", in *IBM Technical Report*, August 1991.
- Tokuda, Hide, "RT-Thread Model for Real-Time Mach", *ITC Workshop*, June 1991.
- Ramos-Thuel, Sandra, Strosnider, Jay K. and Lehoczky, John P., "Performance Impact of Time Redundancy for Backward Err - Recovery in Real-Time Workloads", *Technical Report CMUCDS-91-3, ECE*, April 1991.
- Kirk, D. B., Strosnider, Jay K. and Sasinowski, J. E., "Allocating SMART Cache Segments for Schedulability", in *ONR 3rd Annual Workshop in Foundations of Real-Time Computing*, October 25-26, 1990, Pages 155-168.
- Sathaye, S., Lin A., Bianchini, R. B. and Strosnider, Jay K., "Routing Periodic Real-Time Traffic in a Packet Switched Network", submitted to *12th International Conference on Distributed Systems*, Yokohama, Japan, June 9-12, 1992,) October, 1991.
- Tokuda, Hideyuki and Ashida, Noritake, "rf: An In-memory File System for Real-Time Mach", *ART Project*, July 1991.

- Ishikawa, Y., and Tokuda, H., "Distributed Real-Time Programming Language: RTC++", *Tech. Rpt. in Japanese ONLY, submitted for Journal of Computer Software Japan Society for Software Science Technology, May, 1991.*
- Mraz, R., Palmer, G., Strosnider, Jay K., "Analysis of Architectures for Fault-Tolerant Computation", submitted for *IEEE Transactions on Reliability*, February , 1991.
- Chou, S., Mercer, C. W., Tokuda, H., "Predictable Device Drivers for Real-Time Systems", *January, 1991.*
- Tokuda, H., Mercer, C. W., Ishikawa, Y., Ashida, N., Kimura, N., Chou, S., Breach, S. E., and Poris, M. S., "ARTS System Reference Manual", *Release 1.0, draft*, October, 1990.
- Tokuda, H., Mercer C. W., Ishikawa, Y., Ashida, N., Kimura, N., Chou, S., Breach, S. E., and Poris, M. S., "ARTS User Reference Manual", *Release 1.0, draft*, October, 1990.
- Mercer, C. W., "ARTS Developer Reference Manual", *Release 1.0, draft*, October, 1990.

3.3 Ph.D. Dissertations

- Kirk, David B., "Predictable Cache Design for Real-Time Systems", *Ph.D. Thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University*, November 1990.

Principal Investigator Names:

J. Lehoczky	(412) 268-8725	jpl@k.cs.cmu.edu
L. Sha	(412) 268-5875	lrs@sei.cmu.edu
D. Siewiorek	(412) 268-2570	dps@a.cs.cmu.edu
J. Strosnider	(412) 268-6927	jks@usa.ece.cmu.edu
H. Tokuda	(412) 268-7672	hxt@k.cs.cmu.edu

PI Institution: Carnegie Mellon University
Contract title: Fault Tolerant Real-Time Systems
Contract Number: N00014-84-K-0734
Reporting Period: 1 Oct 90 - 30 Sep 91

4 Transitions and DoD Interactions

ART project personnel frequently interact with DoD representatives, especially Lui Sha in his dual role as a member of the ART project and of the RMARTS project of the CMU-SEI. Dr. Sha is deeply involved with transitioning rate monotonic scheduling theory to industry and government. His efforts include:

- Chairmanship of the real-time task working group of the IEEE Futurebus+ standards committee,
- Interaction with the Navy NGCR,
- Named Chairman of the Board of Visitors of RICIS, an R&D center established by NASA and NASA JSC at University of Houston at Clearlake
- Coordinated the real-time version of POSIX,
- Worked with IEEE 802.6 standards group to develop a real-time capability,
- Continued to develop interpretations of Ada tasking rules to make it suitable for real-time applications.

In addition, Hide Tokuda, as developer of ARTS, interacts regularly with NOSC, IBM and University of Virginia to coordinate the development of testbeds at all four sites and experimentation with ARTS.

Finally, the rate monotonic scheduling theory is increasingly being adopted by major projects. These projects include:

- Navy BSY-1 and BSY-2,
- NASA Space Station Freedom (for system integration)
- European Space Station (recommended its use for its Hard Real-Time OS project).

Principal Investigator Names:

J. Lehoczky	(412) 268-8725	jpl@k.cs.cmu.edu
L. Sha	(412) 268-5875	lrs@sei.cmu.edu
D. Siewiorek	(412) 268-2570	dps@a.cs.cmu.edu
J. Strosnider	(412) 268-6927	jks@usa.ece.cmu.edu
H. Tokuda	(412) 268-7672	hxt@k.cs.cmu.edu

PI Institution: Carnegie Mellon University
Contract title: Fault Tolerant Real-Time Systems
Contract Number: N00014-84-K-0734
Reporting Period: 1 Oct 90 - 30 Sep 91

5 Software and Hardware Prototypes

The ARTS operating system (described in section 2.3 of this report) continues to be developed under the direction of Hide Tokuda and has been distributed to NOSC, IBM, University of Virginia and Cornell. The operating system has an associated tool set, Scheduler 1-2-3 (to determine the schedulability of a task set) and ARM (Advanced Real-Time Monitor). Neither ARTS nor the associated tool set have been commercialized. Hide Tokuda is also developing a real-time version of the Mach operating system, RT-Mach.